Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Absolvování individuální odborné práce

Individual Professional Practice in the Company

2014 **Jiří Svěrák**

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *7. května 2014* …….. ……

podpis studenta

Poděkování

Rád bych poděkoval celé firmě Railsformers, s. r. o., jejímu jednateli Ing. Jiřímu Kubicovi za možnost praxi v jeho firmě vykonat, Ing. Lukáši Kampovi za vedení a hlavně Ing. Zdenku Nevřalovi za velkou pomoc při výběru technologií a následných konzultací při realizaci programu.

Také vedoucí diplomové práce Ing. Zdeňce Chmelíkové, Ph.D. za odbornou pomoc a konzultaci při vytváření této práce.

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

Dne: *7. května 2014* ……..  ……

podpis zástupce

Abstrakt

Bakalářská práce se zaobírá vykonáním mé odborné praxe, již jsem vykonal ve společnosti Railsformers, s. r. o.

Nejdříve uvádím stručný popis společnosti, mé pracovní zařazení a následně použité technologie. Dále popisuji, jak probíhal výběr technologií, analýzu použití pro firmu Railsformers, s. r. o., realizaci zadání ve vybraných technologiích a můj postup. Poté následuje kompletní řešení zadané aplikace, její popis, co vše je ještě třeba implementovat a proč jsem zvolil multiplatformní vývoj. Na konci bakalářské práce uvádím zkušenosti, kterých jsem během ní nabyl, dovednosti, jež mi chyběly, a přínos, který pro mě tato praxe měla.

Klíčová slova

Android; Java; Rhomobile; Ruby; Ruby on Rails; HTML 5; CSS3; Ionic; Javascript; praxe; AngularJS; Railsformers, s. r. o.; AppGyver Steroids

Abstract

In my bachelor´s thesis I am dealing with the topics and issues based on my professional praxis which I have done in the company Railsformers, Ltd.

In the introduction I did a brief description of the company Railsformers, Ltd., my job description there during my praxis and also description of the technologies employed within my work. Consequently I am describing the selection procedure of the technologies, analysis application for company Railsformers, Ltd., performance of the tasks on the selected technologies employed in my work and my workflow. Thereafter the complex solution of the specified aplication follows, its description including all elements which are still needed to be implemented. I also explain there my reasons for choosing a multiplatform solution. In the end of my work I am summarising my experience gained during my praxis, lessons learned and the whole benefit resulting from this work for me.

Key words

Android; Java; Rhomobile; Ruby; Ruby on Rails; HTML 5; CSS3; Ionic; Javascript; praxis; AngularJS; Railsformers, s. r. o.; AppGyver Steroids

Seznam použitých zkratek

Zkratka	Význam
ADT	Android Development Tools
AGS	AppGyver Steroids
AJS	AngularJS
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DI	Dependency Injection
DOM	Document Object Model
HTML	HyperText Markup Language
IDE	Integrated Development Environment
MA	Mobilní Aplikace
MVC	Model-View-Controller
PŘ	Příkazový Řádek
QR	Quick Response
REST	Representational State Transfer
RoR	Ruby on Rails
RM	RhoMobile
Sass	Syntactically Awesome StyleSheets
SEO	Search Engine Optimization
VP	Vývojové Prostředí
WS	WebStorm

Obsah

Úvod

Odbornou praxi jsem vykonával ve firmě Railsformers, s. r. o.

Během ní jsem dostával úlohy, jež se týkaly programování domácího účetnictví pro mobilní telefony s možností synchronizace se serverem, a také které měly potvrdit, jakou technologii následně využívat v celé firmě pro vývoj dalších MA. Z těch jsem si po zkušební aplikaci vybral jednu vyhovující pro další vývoj a v ní naprogramoval zmíněnou aplikaci.

Stručně popíši každou technologii a jazyk. Také popis v nich naprogramovaných aplikací, jejich výhody a nevýhody, a proč jsem nakonec zvolil AJS jako hlavní programovací jazyk a k němu navazující technologie.

1 O firmě a jejím zaměření

Obr. 1.1: *Logo Railsformers, s. r. o.*

Railsformers, s. r. o. je společnost zaměřující se na projekty většího rozsahu, jako jsou enterprise řešení, sociální sítě, SEO optimalizací, komunitními portály a jinými komplexními systémy, ve kterých využívá možností frameworku RoR, využívající výhod týmové spolupráce založené na metodice agilního vývoje.[1]

1.1 Zařazení studenta ve firmě

Ve firmě jsem pracoval na pozici programátor analytik. Mým úkolem bylo rozhodnout se, jaké vývojové frameworky použít, porovnat je na zadané aplikaci a tu pak vyvinout dle zadání.

2 Použité technologie a řešení

Pro vývoj pouze na platformu Android jsem použil oficiálního SDK od společnosti Google Inc., vývojového prostředí IntelliJ IDEA a simulátoru Genymotion.

Multiplatformní vývoj probíhal za pomocí RM, oficiálního nástroje společnosti Motorola Solutions, Inc., postaveného na programu Eclipse. Od něj bylo také upuštěno a konečný vývoj proběhl multiplatformě, použitím frameworku AJS jako programovacího jazyku a WS jako vývojového prostředí.

Pro přehlednost popisuji i RoR.

2.1 Ruby on Rails

Je framework pro vývoj webových aplikací, používající návrhový vzor MVC.

RoR používá jako hlavní programovací jazyk Ruby. Na něm jsou založeny jak šablony vzhledu, tak logika controllerů i architektura aplikace v modelech obalujících databázi.

Mezi základní princip RoR patří „Konvence má přednost před konfigurací“, tedy že programátor konfiguruje pouze ty části aplikace, které se liší od běžného nastavení.[2]

RoR zmiňuji kvůli jeho použití jako serverové části aplikace, jež ale nebyla dokončena a je nutno ji od začátku přepracovat.

2.2 Android SDK

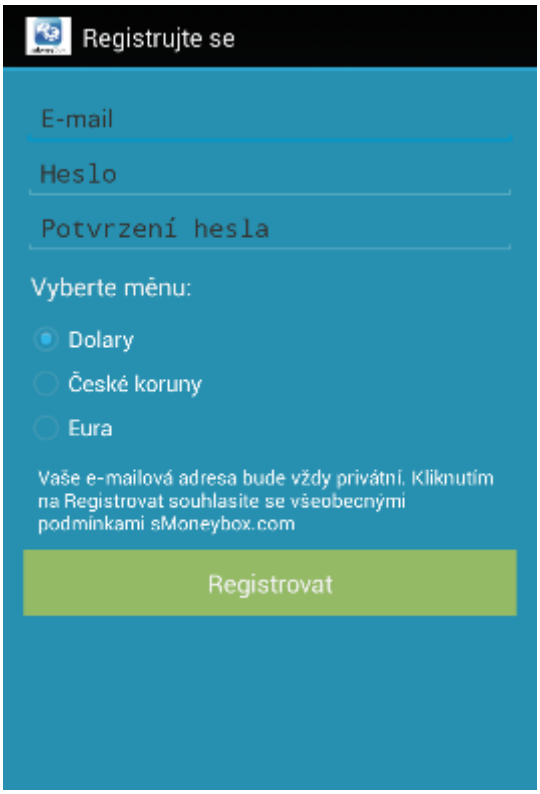
Je kompletní nástroj pro vývoj aplikací na platformu Android. Je dostupný pro tři hlavní platformy, tedy Linux, Windows a Mac OS. SDK sada obsahuje VP Eclipse s ADT pluginem, SDK Tools (nástroje pro ladění, emulátor zařízení, atd.), Platform-tools (nástroje závislé na verzi platformy Android, atd.), poslední verzi platformy Android a obraz nutný pro spuštění emulátoru.

Aktualizace či chybějící a potřebné části lze stáhnout pomocí přibaleného Android SDK Manager.

2.2.1 Aplikace v Android SDK

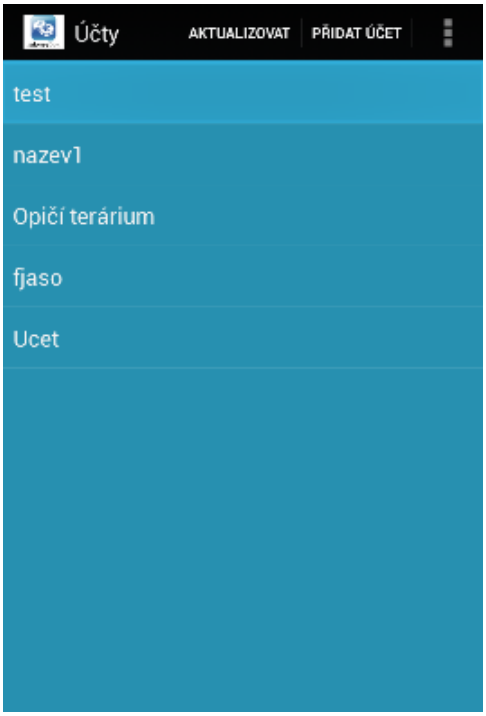
Nativní vývoj pro Android – hlavní platformu, na kterou byla výsledná aplikace cílena – byl zvážen jako první. Rychlost vývoje byla brzděna pouze nehotovým serverovým zázemím, bez něhož jsem nemohl aplikaci zkoušet v reálném čase a případně ladit. Zprvu jsem kvůli jednoduchosti použil REST API.

Přihlášení a registrace byly vyřešeny možností zadat e-mailovou adresu a uživatelské heslo. Po přihlášení je uživateli vygenerován autentizační token. Jeho posílání bylo nutno dodatečně upravit, jelikož nejdříve byl posílán jako GET parametr společně s ostatními daty. Bezpečnostní experti ale zjistili, že tzv. „časováním“, kdy útočník zjišťuje prodlevu mezi kontrolou jednotlivých znaků, lze token zjistit a následně použít k nabourání se do uživatelských dat. Proto byl poté posílán v hlavičce dokumentu.



Obr. 2.1: *Registrace uživatele*

Na Obr. 2.1 lze vidět registraci uživatele i s nastavením měny, jež bylo později provázáno ne s celým uživatelským účtem, ale přidanými bankovními účty. Kurzy měn by se poté stahovaly pomocí jedné z volně přístupných webových API (např. pomocí Google API).



Obr. 2.2: *Výpis účtů*

Na Obr. 2.2 je pouze výpis přidaných účtů, možnost vložit nový, synchronizovat seznam se serverem a další položky v menu.

Výhody nativního vývoje v Android SDK:

Hlavní výhodou je nativní vzhled, na něhož je uživatel zvyklý, a bezproblémový přístup ke všem API Androidu, jako jsou GPS API, Camera API, atd. Také jsem mohl využít svých znalostí Javy a nepotřeboval se učit syntaxi jiného jazyka. Se XML jako značkovacím jazykem pro vzhled aplikace také nebyl problém.

Nevýhody:

Největší nevýhodou pro mě byla zdlouhavost některých řešení a pomalý vývoj, kdy byť jen maličkost měla několik desítek řádků kódu (jako příklad mohu uvést práci s Camera API). Pokud pominu tento subjektivní dojem, největší překážkou byla implementace databáze a následné offline synchronizace. Musel bych v ní totiž řešit i kolize dat ze serveru a MA, což by celý vývoj ještě více zbrzdilo. Nativní vzhled je na jednu stranu velmi dobrý, na druhou hůře modifikovatelný do modernější podoby. U jednodušších aplikací je i nativní programování nevýhodné, jelikož následná implementace řešení na další dvě hlavní mobilní platformy je časově i finančně nákladná oproti vývoji hybridních aplikací. Proto bylo následně od tohoto řešení upuštěno a přesunul jsem se k aplikaci v Rhomobile.

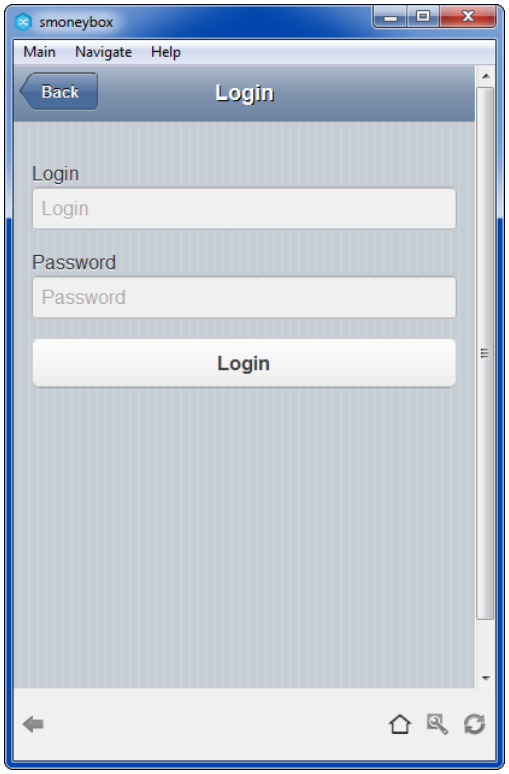
2.3 Rhomobile

RM je framework pro vývoj MA v programovacím jazyku Ruby a je součástí balíku RhoMobile Suite společně s VP RhoStudio (postaveno na VP Eclipse), serverovým RhoConnect a dalšími aplikacemi pro vývoj korporátních aplikací. Hlavním programovacím jazykem je Ruby, vzhled upravitelný pomocí HTML 5 a CSS3.[3]

2.3.1 Aplikace v Rhomobile

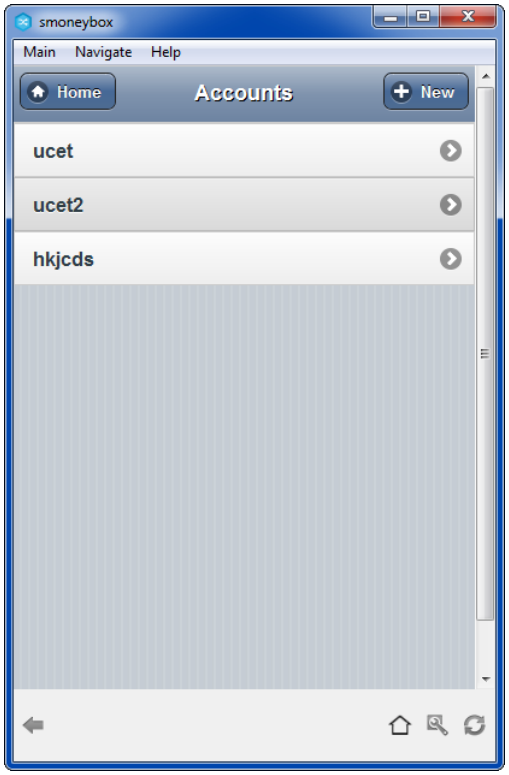
Ze všech zmíněných technologií tato vypadala nejslibněji. Díky letní stáži jsem již uměl Ruby, stejně jako HTML 5 a CSS3. Ruby jsem si okamžitě zamiloval kvůli jeho jednoduchosti a přehledné syntaxi. CRUD operace jsou v RM otázkou několika minut, protože podporuje, po vzoru RoR, generátory v příkazovém řádku. Jeden příkaz vygeneruje celou kostru projektu, druhý zase architekturu MVC pro jeden model se všemi atributy.

Během několika hodin byla aplikace hotova. K jejímu zhotovení mi stačilo pochopit jen některé postupy frameworku, které se lišily od RoR. K tomuto účelu mi posloužila kniha Rhomobile Beginner's Guide, v níž bylo vše vysvětleno a na konkrétních případech i ukázáno.



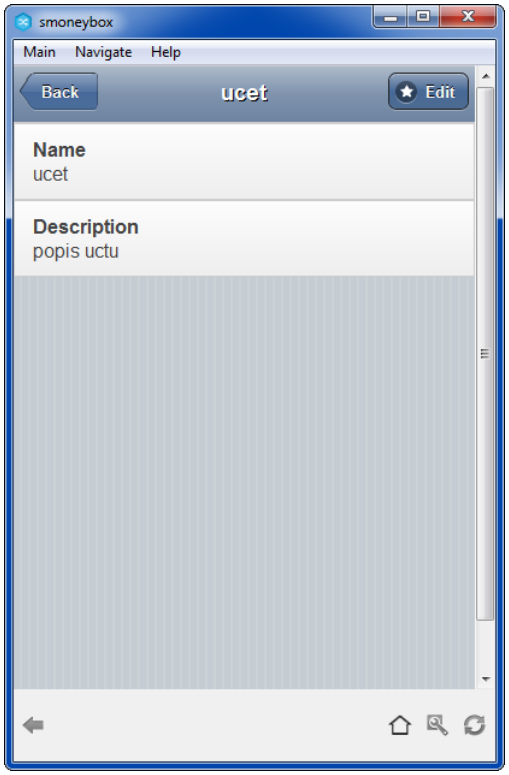
Obr. 2.3: *Přihlášení uživatele*

Obr. 2.3 ukazuje přihlášení uživatele, jež nejdříve komunikuje s aplikací RhoConnect.



Obr. 2.4: *Výpis účtů*

Obr. 2.4 ukazuje výpis všech účtů. Po kliknutí na jeden z nich se zobrazí jeho detail.



Obr. 2.5: *Detail účtu*

Na Obr. 2.5 je zmiňovaný detail účtu. Chybí v něm několik parametrů, které se ale vázaly až k druhé verzi.

První problém vznikl při chybějících vazbách jednotlivých modelů. V knize tato problematika zmíněna nebyla, postup z RoR nefungoval a debugger tak hlásil chybu. Bylo třeba tzv. „natvrdo“ naprogramovat možnost vybrání účtu, jež se váže ke kategorii, což nebyl problém, ale jednalo se o neelegantní a nepoužívané řešení, nicméně bylo nutné.

Druhým a největším problémem bylo propojení se serverem a následující offline synchronizace. Autor knihy doporučil použít součást RhoSuite, RhoConnect. To se jevilo jako elegantní a velmi jednoduché řešení, protože stačilo odkomentovat řádek se synchronizací u každého modelu (vygenerovány společně s modely) a následně několika řádky kódu propojit každý model aplikace s RhoConnect. Meziserver (aplikace RhoConnect) pak řeší konflikty a veškerou komunikaci mezi serverem a klientskou aplikací. Bohužel je omezena na deset připojených zařízení – po připojení jednoho klienta bylo možné spojit již jen devět dalších. Toto omezení vyplývá z omezení RhoConnect. Pro vývojový režim je povoleno použití pouze deseti zařízení (na testování). Pokud se aplikace vypustí do produkčního prostředí, je třeba zakoupit licenci. Motorola neuvádí žádný oficiální ceník, nýbrž zmínku o možnosti kontaktovat prodejní oddělení, které následně, po sdělení plánovaného měsíčního provozu – počtu připojených zařízení –, zpětně zašle cenovou nabídku. Napsal jsem tedy prodejnímu oddělení, jaký mi nabídnou měsíční tarif při plánovaném provozu do sta připojených klientů. E-mail zůstal nezodpovězen. Po týdnu jsem psal znovu e-mail podobný tomu prvnímu, na který jsem se

odpovědi také nedočkal. Další kontakt již nepřipadal v úvahu. Orientační ceník, který poskytl jeden z tvůrců frameworku na Google Groups, uváděl tak přemrštěnou částku, že bylo toto řešení synchronizace okamžitě zavrhnuto. To byla veliká škoda, jelikož se již aplikace mohla posunout směrem k produkčnímu prostředí.

Výhody multiplatformního vývoje v Rhomobile:

V první řadě je to rychlost, s jakou framework generuje modely s CRUD operacemi. Téměř vše vyřeší za programátora. Při tom se drží zásad RoR, kdy má konvence přednost před konfigurací.

Vzhled aplikace je možné upravit pomocí HTML5 a CSS3. Pokud se programátor spokojí s již definovanými vzhledy, není třeba do nich zasahovat.

Databázi řeší ve výchozím stavu mapperem Property Bag. Pro rozsáhlejší tabulky je již nutno použít Fixed Schema, které je sice ve výsledku mnohem menší (tedy velikost celé databáze), ale při změně atributů je potřeba řešit migrace. Opět se jedná o velmi elegantní a jednoduché řešení.

Zabudovaný emulátor je výhodou, jelikož není závislý na Android SDK.

RM podporuje mnoho platforem – Android 2.3 a výše, iOS 6.0 a výše, BlackBerry, Windows Phone 8, Windows Mobile 5 a výše, atd.

Nevýhody:

Vazby databáze zcela nepochopitelně chybí, i když se jedná o velmi důležitou komponentu. Lze je sice částečně doimplementovat, nicméně doufám, že v dalších verzích frameworku již chybět nebudou.

Spojení serveru a klientské aplikace řeší opět jednoduše RhoConnect. Bohužel dle jeho ceny lze usuzovat, že se jedná o korporátní aplikaci, tedy že se pro menší firmy a programy nehodí. Také přístup celého týmu kolem RM je mi záhadou, jelikož nejsou ochotni ani odpovědět potencionálnímu zákazníkovi.

S předchozí výtkou souvisí i neaktuálnost verze Ruby, na jejíž aktualizaci je třeba počkat i několik měsíců.

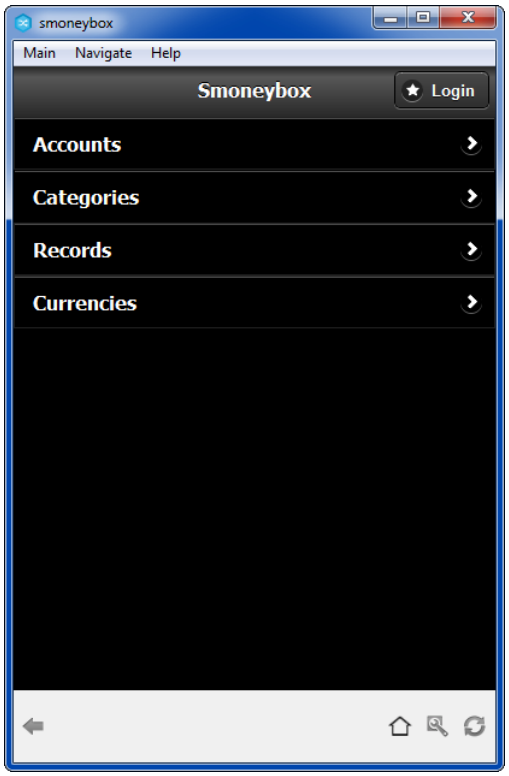
Nezájem vývojářů frameworku je vidno i u zastaralé dokumentace a orientace v ní, kdy se mísí dokumentace staré verze s novou.

Bohužel nelze pro vývoj využít obrovské výhody RoR – gemů. Gem je, zjednodušeně řečeno, něco jako knihovny v Javě. Je jich velké množství a jejich implementace je velmi snadná. Přesněji řečeno, použití není podporováno, ale některé gemy, jež jsou naprogramovány čistě v Ruby (tedy bez podpory knihoven z jazyka C či Javy), použít lze. Těch je bohužel minoritní počet.

Při přechodu z jiného frameworku či programovacího jazyka předpokládáme alespoň malou podporu vývojářské komunity. Ta je u RM bohužel velmi malá a soustředí se jen kolem

několika málo programátorů. Mé tvrzení dokládá i množství knih o vývoji v RM – existují pouze dvě.

Výchozí vzhledy aplikace jsou sice mnohdy dostačující, ale pokud má být aplikace alespoň trochu modernější, je třeba je změnit pro každou platformu.



Obr. 2.6: Ukázka vzhledu v Androidu

Na Obr. 7 lze vidět výchozí vzhled programu na platformě Android.

Od verze RM 4.0.0 je možné vyvíjet aplikace pouze pod Windows, což je pro mne také velmi důležitá negativní vlastnost.

K předchozímu bodu se váže i nutnost využívat ofic. IDE – RhoStudio –, jež má jako základ IDE Eclipse. Tato výtka je objektivní, jelikož je celé prostředí nestabilní, bylo nutné jej mnohokrát restartovat (nebo rovnou celý počítač), hlásí chyby programu i tam, kde nejsou, atd.

Stále jsem přesvědčen, že je RM velmi dobrý framework, nehledě na vyjmenované nedostatky. Doufám, že je v budoucnu vyřeší změna majitele či výměna vedení.

2.4 AppGyver Steroids, AngularJS a AngularFire

Appgyver Steroids je nástavba nad frameworkem PhoneGap, s nímž sdílí ovládání skrze příkazový řádek, jádro systému a API jednotlivých platforem. Rozdílný je přístup ke vzhledu

výsledné aplikace a další možnosti, jako je generování QR kódu (příkaz v PŘ spustí emulátor a vygeneruje QR kód), které umožňuje aplikaci okamžitě vyzkoušet na jakémkoli mobilním telefonu. V současné době se stává stále populárnějším právě díky přístupu k debuggování, množství doplňků, zkoušení reálné aplikace a možnosti odeslat QR kód komukoli, kdo dříve nainstaloval aplikaci AppGyver Scanner (je k dispozici na třech hlavních mob. platformách) a potřebuje aplikaci také vyzkoušet – klient či nadřízený programátora.[4]

AngularJS je javascriptový MVC framework společnosti Google Inc., který rozšiřuje HTML dokument o řadu elementů a atributů. Umožňuje vytvářet vlastní HTML elementy a atributy, obsahuje dvoucestné provázání – změna v HTML se okamžitě projeví i v jeho controlleru. Manipulace s HTML je tedy mnohem jednodušší a přehledná.[5] Pro vývoj jsem nepoužil čistý AngularJS, ale nástavbu Ionic, která kombinuje moderní vzhled a přidává k AngularJS mnoho dalších DI a podporuje i hardwarovou akceleraci animací a přechodů, jež se následně i na pomalejším zařízení nezasekávají.

Firebase je velmi mocný doplněk pro ukládání dat, kontakt s meziserverem a následnou synchronizací klientských i serverových aplikací. AngularFire je pak Firebase pro použití s AngularJS, který rozšiřuje o trojcestné provázání, tedy že jakákoli změna na jednom zařízení se okamžitě projeví i na ostatních. Jako příklad mohu uvést fungování chatu. Jeho síla také spočívá v automatickém offline ukládání dat, která synchronizuje až při připojení zařízení k internetu. Dalšími výhodami je jeho doplněk Firebase Simple Login, jež umožňuje registraci a přihlášení uživatelů.[6]

2.4.1 Řešení aplikace v AppGyver Steroids a AngularJS

Po dlouhém hledání jsem konečně našel nejpohodlnější způsob, jak vyvíjet mobilní aplikace. Neustále vzrůstající obliba těchto technologií u vývojářů mi dala za pravdu, že jsem vybral dobře. Také se vyplatilo hledání dalších doplňků – knihoven –, jež mi pomohly splnit zadání přesně podle požadavků zadávajícího.

Nejprve jsem si vygeneroval nový projekt příkazem

```
$ steroids create smoneybox
```

Poté jsem mohl ihned začít programovat.

Příkazem

```
$ steroids connect --serve
```

se pouští webový server a generátor QR kódu.

2.4.2 Cesty aplikace

Pro přehlednost a kvůli zjednodušení neuvádím kompletní zdrojové kódy. Následující kód zobrazuje cesty aplikace:

```
app.config(['$stateProvider', '$urlRouterProvider',
function($stateProvider, $urlRouterProvider) {
    $stateProvider

        .state('index', {
            url: '/index',
            controller  : 'MainCtrl'
        })

     ...

        .state('settings', {
            url: '/settings',
            templateUrl : '/views/settings.html',
            controller  : 'SettingsCtrl'
        });
    $urlRouterProvider.otherwise('/index');
}]);
```

2.4.3 Factory

Factory je speciální funkce (poskytovatel), která sdílí data mezi jednotlivými controllery:

```
app.factory("DataFactory", function($firebase){
    return{
        accounts:
            accounts = $firebase(accountsFire)
        ,
        categories:
            categories = $firebase(categoriesFire)
```

```
        ,
        records:
            records = $firebase(recordsFire)
        ,
        currencies:
            currencies = $firebase(currenciesFire)
        ,
        users:
            users = $firebase(usersFire)
    }
});
```

2.4.4 Dependency injection

Zde jsou předány jako parametry funkce tzv. dependency injection, jež říkají, jak jsou jednotlivé komponenty provázány s jinými závislostmi. Jako závislost je předána i výše zmiňovaná factory – DataFactory:

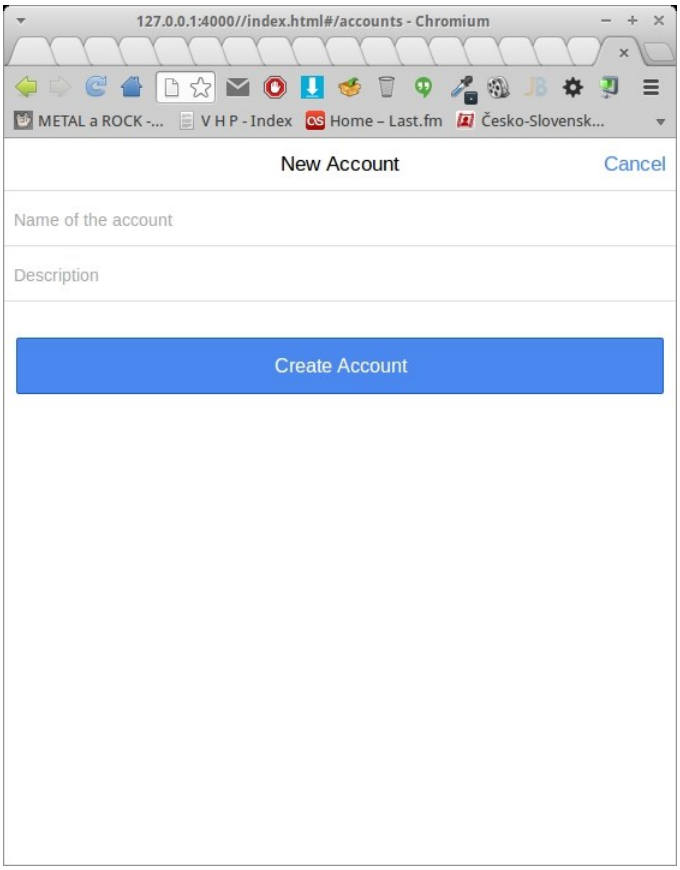
```
app.controller('AccountCtrl', function($scope, $ionicModal,
$firebase, DataFactory) {
...
});
```

2.4.5 Vytvoření účtu

Metoda, jež má na starosti vytvoření nového účtu a jeho zaslání serveru. Scope je zvláštní případ objektu, do kterého lze uložit ostatní objekty a je také jakýmsi „lepidlem“ mezi view a controllerem:

```
$scope.createAccount = function(account) {
        $scope.accounts.$add({
            name: account.name,
            description: account.description,
            currency: account.currency
        });
        $scope.closeNewAccount();
```

```
        account.name = "";
        account.description = "";};
```



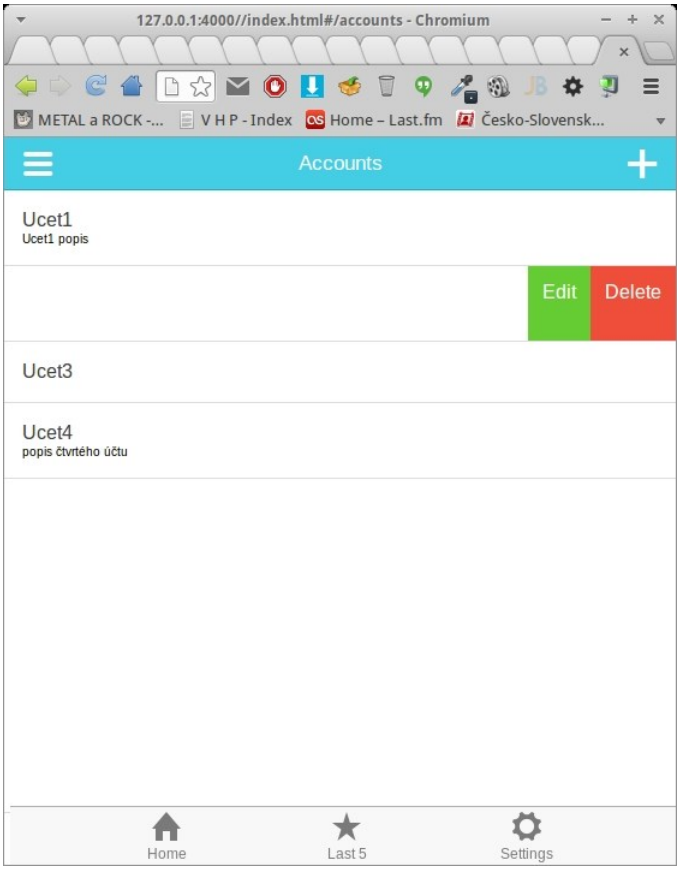
Obr. 2.6: *Nový účet*

Na Obr. 2.6 lze vidět vytvoření nového účtu. Zašedlé písmo znázorňuje nápovědu uživateli.

2.4.6 Smazání účtu

Tato metoda smaže účet z MA i ze serveru. Pokud není server dostupný (zařízení není připojeno k internetu), akce se provede automaticky při prvním připojení. Metodě remove je potřeba předat klíč účtu, jelikož ukládání dat na Firebase serverech probíhá právě podle vygenerovaných klíčů:

```
$scope.deleteAccount = function(key) {
        $scope.accounts.$remove(key);
    };
```



Obr. 2.7: Smazání a editace účtu

Obr. 2.7 ukazuje řešení změny či smazání účtu. Nabídka se zobrazí posunutím účtu doleva.

2.4.7 Registrace uživatele

Metoda registrace uživatele. Je vázána ještě k proměnné auth, která definuje třídu FirebaseSimpleLogin. Jak lze vyčíst, veškeré operace jsou jednoduché a dobře čitelné. Pro přehlednost a kvůli účelu testování vypisuje do konzole i citlivé informace jako heslo uživatele:

```
$scope.newUser = function(user){
        $scope.userEmail = user.email;
        $scope.userPassword = user.password;

        auth.createUser($scope.userEmail, $scope.userPassword,
function(error, user) {
            if (!error) {
```

```
                console.log('User UID: ' + user.uid + ', Email:
' + $scope.userEmail + ', User ID: ' + user.id + ', User
Password: ' + $scope.userPassword);
            }
            $scope.userId = user.id;
            $scope.userUid = user.uid;
            $scope.addUserToFirebase($scope.userEmail,
$scope.userUid, $scope.userId);
        });
        $scope.registerModalHide();
        user.email = '';
        user.password = '';
    };
```

Výhody a důvod výběru AngularJS a dalších:

V žebříčku nejpoužívanějších javascriptů framework AngularJS právem vede. S HTML pracuje způsobem, který významně snižuje náročnost psaného kódu a počet jeho řádků (uvádí se, že až na 10 % oproti kódu v Javě).[7]

Počet diskuzních fór, skupinových diskuzí a množství dotazů na Stack Overflow ukazuje, jak je komunita kolem AngularJS rozšířená, a to proto, že se dá využít k oživení webových stránek, i naprogramování kompletní aplikace pro mobilní telefony.

CSS styly lze velmi jednoduše přepisovat, protože Ionic podporuje Sass.

Ionic přidává mnoho dalších funkcionalit, které programování v AngularJS dále zjednodušují, a navíc přidává moderní vzhled i několik set ikon. Je vytvořen na míru programování MA. Proto tak zásadně zjednodušuje už tak skvělý framework, jakým je AngularJS.

3 Teoretické a praktické znalosti a dovednosti, které jsem získal v průběhu studia a uplatnil v průběhu odborné praxe

Následující předměty mi pomohly k vykonání praxe:

Algoritmy II – Tento předmět mě uvedl do problematiky objektově orientovaného programování.

Programovací jazyky I – Předmět mě naučil Javu, kterou jsem využil nejen v dalším ročníku, ale především pro naprogramování nativní aplikace, i když jsem ji nakonec nevyužil.

Tvorba aplikací pro mobilní zařízení I – Předmět mi ukázal, jak vznikají MA, přestože se jednalo o již nepoužívanou platformu.

Tvorba aplikací pro mobilní zařízení II – Tohoto předmětu jsem využil nejvíce, jelikož mne zasvětil do vývoje Android aplikací.

4 Znalosti a dovednosti, které mi chyběly v průběhu odborné praxe

V průběhu praxe mi chyběly hlavně hlubší znalosti programování a také to, jaký framework nejlépe použít pro konkrétní zadání. Dále mezery v programování pro Android a mnou dlouho nepoužívaném jazyku – Ruby. V javascriptu jsem do té doby neuměl programovat vůbec, natož abych použil nějakou jeho nástavbu. Nakonec se ale zařadil mezi mé oblíbené programovací jazyky po boku Javy a Ruby.

5 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Odborná praxe mne naučila, jak společnost vybírá co nejlepší nástroje pro zadané řešení, s těmi jsem se následně naučil velmi dobře pracovat. Do značné míry jsem prohloubil znalosti programování a to ve více jazycích zároveň.

Také spolupráci v týmu, kdy řešení není problémem jednotlivce, ale celé skupiny.

6 Závěr

Celá praxe pro mě měla velký přínos, dělal jsem v týmu, konzultoval svá řešení s vedením a naučil se mnoho zajímavých postupů.

Naučil jsem se řešit problémy pod tlakem a vymýšlet zlepšení dané aplikace oproti zadání. Výsledná aplikace bohužel není ihned použitelná, jelikož serverová část není hotova a musí se kompletně přepracovat, nicméně aplikace funguje velmi dobře i bez ní.

Použitá literatura

[1] Railsformers.com. *Railsformers.com* [online]. © 2014 [cit. 2014-05-04]. Dostupné z: http://www.railsformers.com/cs

[2] Ruby on Rails. *RubyOnRails.cz* [online]. 2004 [cit. 2014-05-04]. Dostupné z: http://www.rubyonrails.cz

[3] RhoMobile Suite - Motorola Solutions USA. MOTOROLA SOLUTIONS. *RHOMOBILE SUITE* [online]. © 2014 [cit. 2014-05-04]. Dostupné z: http://www.motorolasolutions.com/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite#

[4] Create mobile apps with native performance using HTML5. APPGYVER, INC. *AppGyver* [online]. © 2010-2013 [cit. 2014-05-04]. Dostupné z: http://www.appgyver.com/

[5] HTML enhanced for web apps!. GOOGLE, INC. *AngularJS* [online]. ©2010-2014 [cit. 2014-05-04]. Dostupné z: https://www.angularjs.org/

[6] Build Realtime Apps. FIREBASE, Inc. *Firebase* [online]. 2014 [cit. 2014-05-04]. Dostupné z: https://www.firebase.com

[7] Začínáme s AngularJS. DEVEL.CZ LAB S.R.O. *Zdrojak.cz* [online]. 2014 [cit. 2014-05-04]. Dostupné z: http://www.zdrojak.cz/clanky/zaciname-s-angularjs/

Seznam příloh

Součástí BP je CD.

Adresářová struktura přiloženého CD:

/AndroidSDK – aplikace v Android SDK

/Rhomobile – aplikace v Rhomobile

/Steroids – aplikace v AppGyver Steroids a AngularJS